



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/014,476	12/14/2001	Aleksandr O. Ryzhov	10018532-1	1020

7590 09/24/2004

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER

STEELMAN, MARY J

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 09/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/014,476

Applicant(s)

RYZHOV, ALEKSANDR O.

Examiner

Mary J. Steelman

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12/14/2001, 1/13/2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 14 December 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 01/13/2004.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 are pending.

Information Disclosure Statement

2. IDS received 14 January 2004 has been considered.

Drawings

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: FIG. 3 #360, FIG. 4, #422 (See page 7, line 4, #442 should be corrected to be #422). Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The use of the trademark JAVA has been noted in this application. It should be capitalized (including in the Drawings and Abstract) wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 1 - 20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

7.35.01 Trademark or Trade Name as a Limitation in the Claim

Claims 1, 5, 6, 13, 17 and 20 contain the trademark/trade name JAVA. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe byte code programming language and, accordingly, the identification/description is indefinite.

For compact prosecution Examiner is treating the limitation as if it reads on "byte code programming language."

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,275,978 B1 to Bell, in view of US Patent 6,584,612 B1 to Mueller et al.

Per claims 1 and 17:

Bell disclosed (col. 1, line 65-col. 2, line 38) localization using resource bundles and a key to assist in locating a particular term or string. Col. 6, lines 18-22, "During execution of the object program, the resource bundle program file, residing within the object code, is accessed to provide term (i.e., messages) localization and, if required, term differentiation localization in cases of terms having multiple meanings."

Bell disclosed:

-locating a plurality of localizable variables...

(Col. 2, lines 21-23, "The resource bundle program file contains a localization term library (plurality of localizable variables), and the resource program file is a source file for loading the resource bundle program file.")

Art Unit: 2122

-finding a corresponding resource file for a current language for each localizable variable;

(Col. 2, lines 28-29, "...maintaining a list of a plurality of localized differentiated terms...")

-calculating a key for each localizable variable;

(Col. 4, lines 8-10, "The resource bundle program file is a library of localization differentiation terms that include at least a first key...")

-finding a localized string in the resource file corresponding to each key;

(See FIGs. 2 & 3.)

-assigning the localized string to the corresponding localizable variable of the class,

(Col. 6, lines 3-6, "The resource bundle generator files built to assist in the localization are the resource program file which is a source file for loading the localization term library created in the resource bundle program file..." The localization term library assigns the localized string to the corresponding localizable variable of the class.)

- provides localization of the class during class loading.

(Col. 5, lines 35-43, "The program compiler creates the object code (classes are loaded during compiling) with the incorporated resource bundle localization terms (localization of the class)...")

Bell failed to disclose details related to the loading of classes from the resource file. However Mueller provided more details regarding a custom class loader adapted to load resource files.

Art Unit: 2122

Mueller disclosed:

-a class using a custom class loader;

(Col. 3, lines 58-64, "...executing a ROM class loader on the JVM, the ROM class loader being adapted to loading the resource files from ROM...receiving by the ROM class loader, a request from the application program...", col. 6, lines 64-67, "...an object class referred to as 'RomLoadingClassLoader', which is a novel JAVA class loader that knows how to load resources out of ROM...", col. 8, lines 56-58, "Techniques for specifying a class loader as an option of the JVM are known in the art...")

Sub-classing the primordial class loader is defined in the JAVA programming language. A sub-classed loader can be adapted to load specified files from specified locations.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Bell's invention, to include details related to the loading of classes using a custom class loader because it is a well known feature of the JAVA programming language, suitable for redirecting an executing program.

Per claims 2 and 18:

-creating an instance of the custom class loader using a launcher;

(Col. 2, lines 16-23, "...system can be broadly summarized by the following components; program source code, a resource bundle system that interacts with the program source code, verifies compliance of the program source code with the syntactical rules, and generates data

Art Unit: 2122

files to assist in the localization of source data values. The resource bundle program file contains a localization term library, and the resource program file is a source file for loading (create custom class loader) the resource bundle program file”, col. 3, lines 43-44, “...the resource program file (custom class loader) is a source file for loading the resource bundle program file ...”, col. 3, lines 46-52, “The resource bundle file and resource program file are provided as inputs, along with the program source code to the program compiler. The program compiler generates a program object code. The program object code includes an object code version of both resource program file (custom class loader) and resource bundle program file for execution of the term localization differentiation.” The program compiler generates a custom class loader that diverts loading to the resource bundle.)

-requesting an application startup class from the custom class loader;

(Col. 4, lines 2-7, “The resource bundle generator files built to assist in the localization differentiation include, a resource bundle file, which is a localization differentiation term library, and a resource program file, which is a source file for loading (application startup class) the localization differentiation term library resource bundle program file. col. 4, lines 11-17, “Upon completion of the execution of the generate resource bundle system, the resource bundle file and resource program file are provided as inputs, along with the program source code to the program compiler. The program compiler generates a program object executed by the processor.” The custom class loader is used to start up the application.)

-returning a localized version of the application startup class;

Art Unit: 2122

(Col. 4, lines 46-51, “The resource program file 80 and resource bundle program file 90 are embedded within the program object code as resource program file 80’ and resource bundle program file 90’ to provide term localization differentiation during execution of the program object code.” The localized version of the application startup class is in 80’.)

-running the application startup class using a reflection application programming interface (API). (Col. 6, lines 18-22, “During execution of the object program, the resource bundle program file, residing within the object code, is accessed to provide term (i.e., messages) localization...” The Reflection API is defined in the JAVA programming language. It supports dynamic retrieval of information about classes and allows manipulation with an executing JAVA program.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to use the reflection API to access the resource bundle to manipulate the executing program by substituting a localized version of a string.

Per claims 3 and 19:

-loading resource files of the class from a storage device using the custom class loader.

(Col. 4, lines 11-17, “Upon completion of the execution of the generate resource bundle system, the resource bundle file and resource program file are provided as inputs, along with the program source code, to the program compiler code, which can be a machine code that can be directly executed (loaded resources files used for execution) by the processor.”

Per claims 4 and 20:

Art Unit: 2122

-loading the class from a bootstrap class loader;

(Col. 3, lines 38-41, "The generate resource bundle system performs a precompilation (load class from a bootstrap class loader) of source program and creates files that assist in the localization of terms within the source program.")

-passing a localized class to a JAVA virtual machine.

(Col. 2, lines 32-38, "...providing a differentiation token of said token, generating a localization token...and generating object code (localized class) of said source code including said localization token." The Bell reference is associated with the JAVA programming language (col. 1, lines 25-29). It is well known in the art, that a virtual machine is used to process the language.)

Per claim 5:

-a custom class loader capable of inspecting codes of the application, and converting strings of a class to localized strings during class loading, thereby providing localization of the class during class loading;

(Col. 4, lines 17-21, "The resource program file and resource bundle program file are embedded within the program object code as resource program file 80' and resource bundle program file 90' to provide term localization differentiation during execution of the program object code."

The embedded code causes an executing program to use custom class loaders to load from the library of localization differentiation terms using a key.)

-a reflection application programming interface (API) capable of obtaining descriptors of variables of the class;

(Col. 6, lines 18-22, "During execution of the object program, the resource bundle program file, residing within the object code, is accessed to provide term (i.e., messages) localization..." The Reflection API is defined in the JAVA programming language. It supports dynamic retrieval of information about classes and allows manipulation with an executing JAVA program.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to use the reflection API to access the resource bundle to manipulate the executing program by substituting a localized version of a string.

-a launcher capable of creating an instance of the custom class loader and running an application startup class using the reflection API.

(Col. 4, lines 11-17, " , col. 6, lines 18-22, "During execution of the object program, the resource bundle program file, residing within the object code, is accessed to provide term (i.e., messages) localization..." The Reflection API is defined in the JAVA programming language. It supports dynamic retrieval of information about classes and allows manipulation with an executing JAVA program.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to use the reflection API to access the resource bundle to manipulate the executing program by substituting a localized version of a string.

Per claim 6:

Art Unit: 2122

-the custom class loader is a subclass of a JAVA language class loader.

(Col. 1, lines 45-47, "At runtime the programmer uses the 'ResourceBundle.getBundle()' method to load (custom loaders) the appropriate 'Resource Bundle' class for the current locale", col. 4, lines 23-25, "localization differentiation system of the present invention utilizes the resource bundle functionality provided by the JAVA system...")

Regarding claims 7-11, see the rejection of limitations in claim 1 above.

Regarding claims 12 and 13, see the rejection of limitations in claim 4 above.

Regarding claims 14-16, see the rejection of limitations in claim 2 above.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2122

After October 25, 2004, examiner can be reached at new telephone number (571) 272-3704. Supervisor, Tuan Q. Dam can be reached at (571) 272-3694.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman



09/16/2004



TUAN DAM
SUPERVISORY PATENT EXAMINER